

ИСП 1.1

Выполнил: Нахатакян Артур, ИВТ 3 курс

Задача: провести инсталляцию программного обеспечения.

На основе материалов, опубликованных в примечаниях к заданию, установить необходимое ПО. В отчете поэтапно продемонстрировать процесс установки (в виде набора скриншотов) с ключевыми моментами установки.

Примечание 1: <https://disk.yandex.ru/i/bHPfXyufLTK0og>

Выполнение:

1. Создаем директорию с нашим проектом и копируем туда *Dockerfile* с <https://github.com/getgrav/docker-grav/blob/master/Dockerfile>
2. Создаем *docker-compose.yml* и копируем в него содержимое, которое находится в самом конце *README.md* файла <https://github.com/getgrav/docker-grav/blob/master/README.md>

Running Grav Image with docker-compose and a volume mapped to a local directory

Running `docker-compose up -d` with the following docker-compose configuration will automatically build the Grav image (if the Dockerfile is in the same directory as the docker-compose.yml file). Then the Grav container will be started with all of the site data persisted to a named volume (stored in the `./grav` directory).

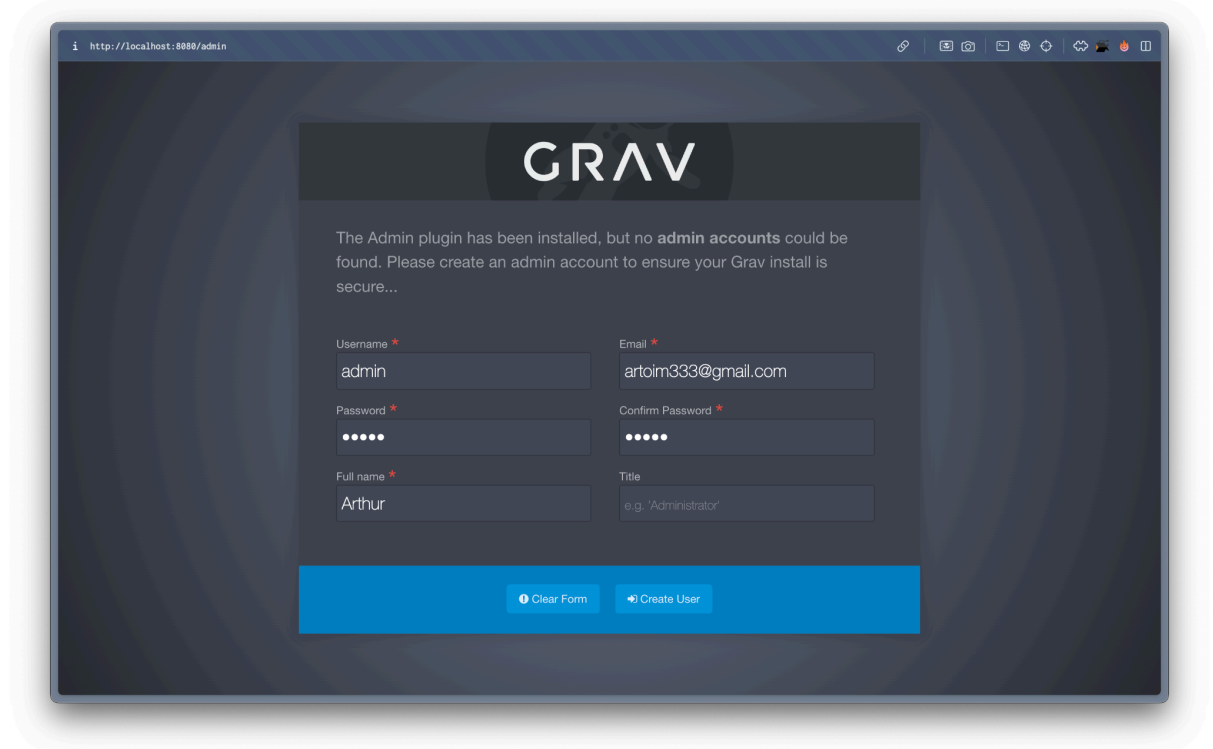
```
volumes:
  grav-data:
    driver: local
    driver_opts:
      type: none
      device: $PWD/grav
      o: bind

services:
  grav:
    build: ./
    ports:
      - 8080:80
    volumes:
      - grav-data:/var/www/html
```

3. Далее выполняем `docker-compose up -d`, после чего запустится контейнер с нашим приложением.

```
➔ grav-test docker-compose up -d
Compose now can delegate build to bake for better performances
Just set COMPOSE_BAKE=true
[+] Building 1.3s (15/15) FINISHED
=> [grav internal] load build definition from Dockerfile
=> => transferring dockerfile: 2.40kB
=> [grav internal] load metadata for docker.io/library/php:8.3-apache-bullseye
=> [grav auth] library/php:pull token for registry-1.docker.io
=> [grav internal] load .dockerignore
=> => transferring context: 2B
=> [grav 1/9] FROM docker.io/library/php:8.3-apache-bullseye@sha256:6415657dc5b12c6da256716eeeff9e5ac3a373d37def57ccd162e0
=> CACHED [grav 2/9] RUN a2enmod rewrite expires && sed -i 's/ServerTokens OS/ServerTokens ProductOnly/g' /etc/apac
=> CACHED [grav 3/9] RUN apt-get update && apt-get install -y --no-install-recommends unzip libfreetype6-dev l
=> CACHED [grav 4/9] RUN { echo 'opcache.memory_consumption=128'; echo 'opcache.interned_strings_buffer=8'; ec
=> CACHED [grav 5/9] RUN pecl install apcu && pecl install yaml-2.2.4 && docker-php-ext-enable apcu yaml
=> CACHED [grav 6/9] RUN chown www-data:www-data /var/www
=> CACHED [grav 7/9] WORKDIR /var/www
=> CACHED [grav 8/9] RUN curl -o grav-admin.zip -SL https://getgrav.org/download/core/grav-admin/latest && unzip grav-
=> CACHED [grav 9/9] RUN (crontab -l; echo "* * * * * cd /var/www/html;/usr/local/bin/php bin/grav scheduler 1>> /dev/null
=> [grav] exporting to image
=> => exporting layers
=> => writing image sha256:67afc9aa1c1ecda2ff0225567d45fd85864ed962dbc6a280eb3b3b2622459c28
=> => naming to docker.io/library/grav-test-grav
=> [grav] resolving provenance for metadata file
[+] Running 4/4
✔ grav Built 0.0s
✔ Network grav-test_default Created 0.1s
✔ Volume "grav-test-grav-data" Created 0.0s
✔ Container grav-test-grav-1 Started 0.3s
➔ grav-test
```

4. Далее открываем <http://localhost:8080>, где откроется панель создания юзера. (Если на данном порту не открылся grav, можно воспользоваться командой `docker ps` и посмотреть на каком порту запустилось приложение)



5. Вводим наши данные и нажимаем на создание юзера, после чего откроется админ панель Grav. Добро пожаловать!

